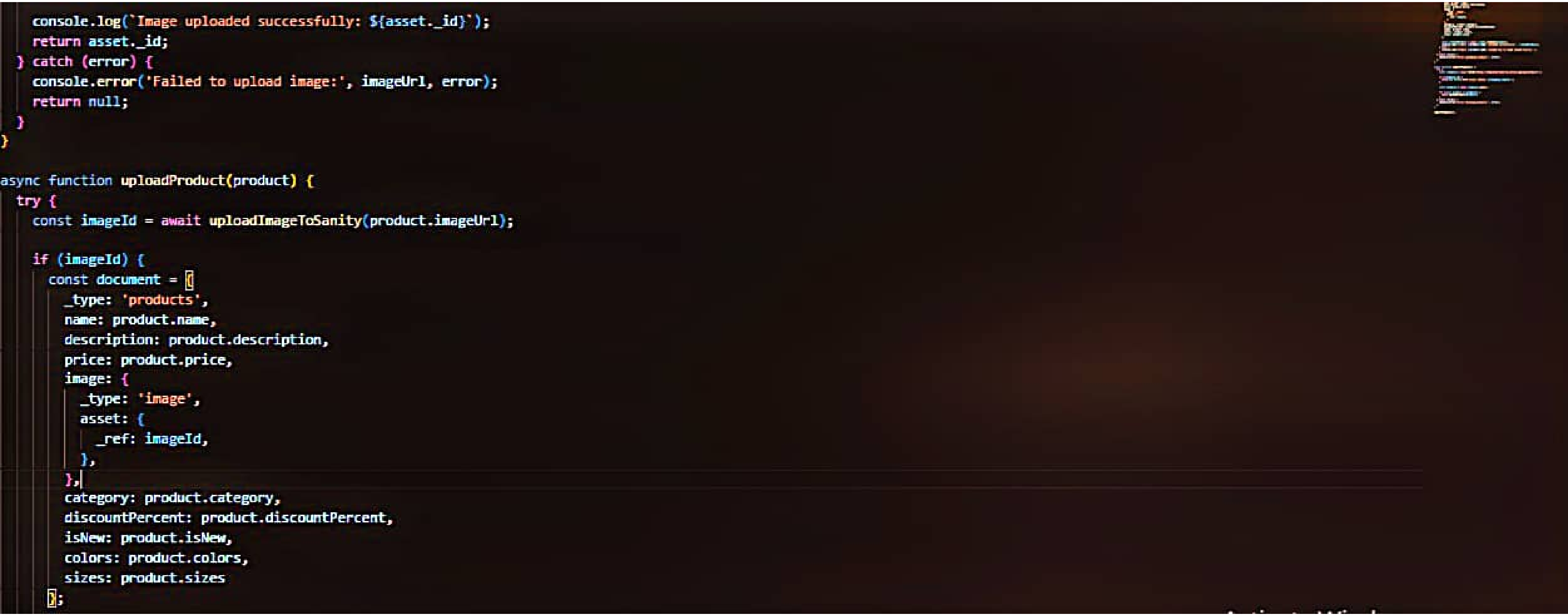
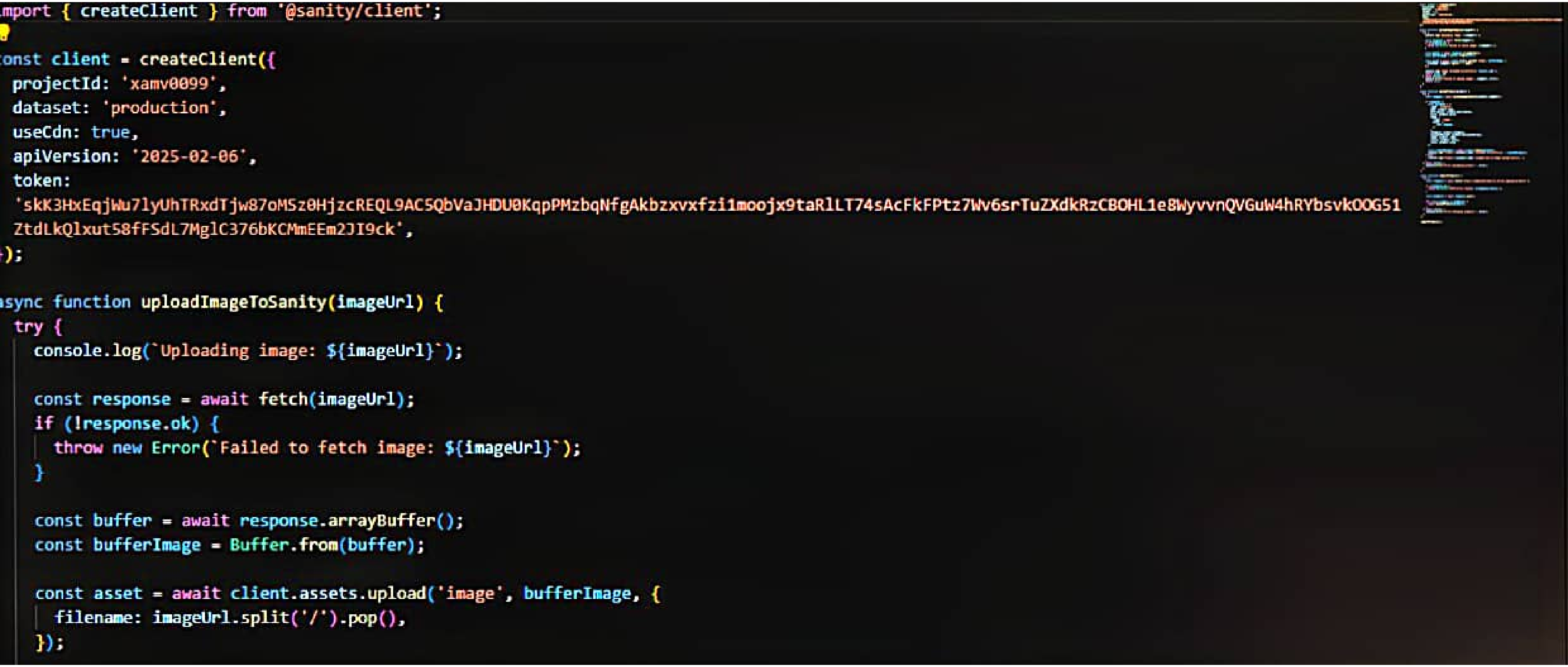
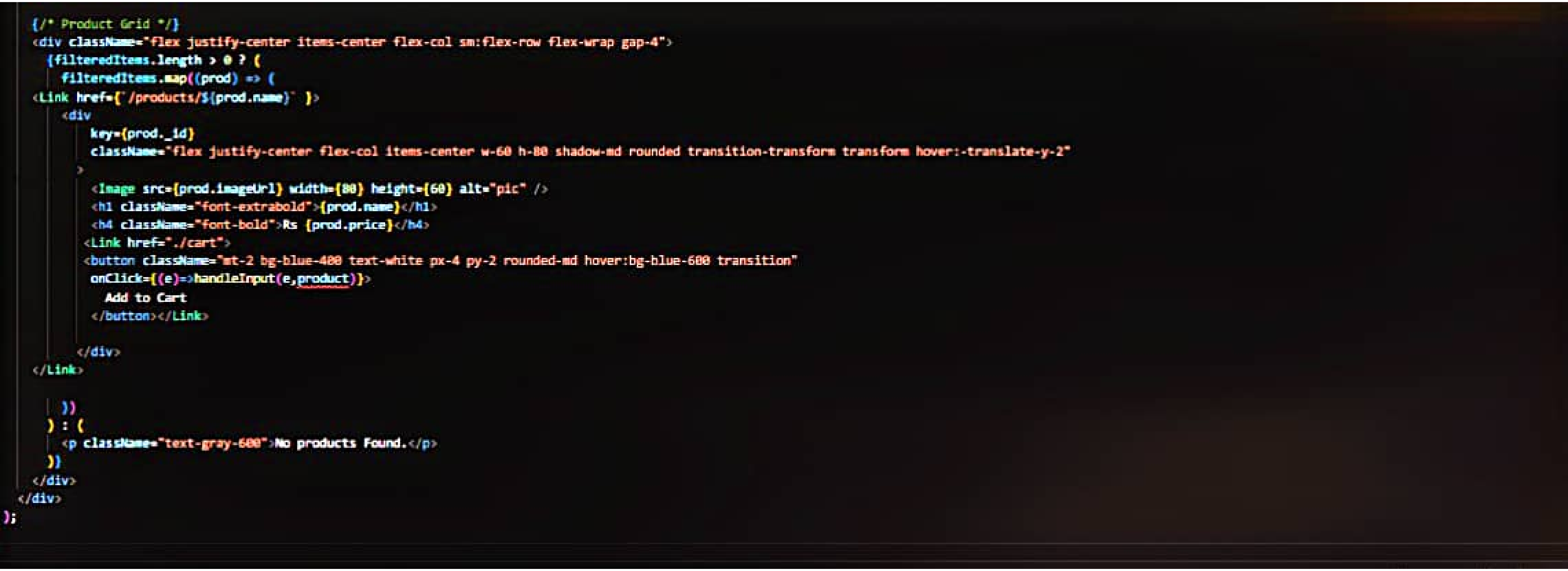


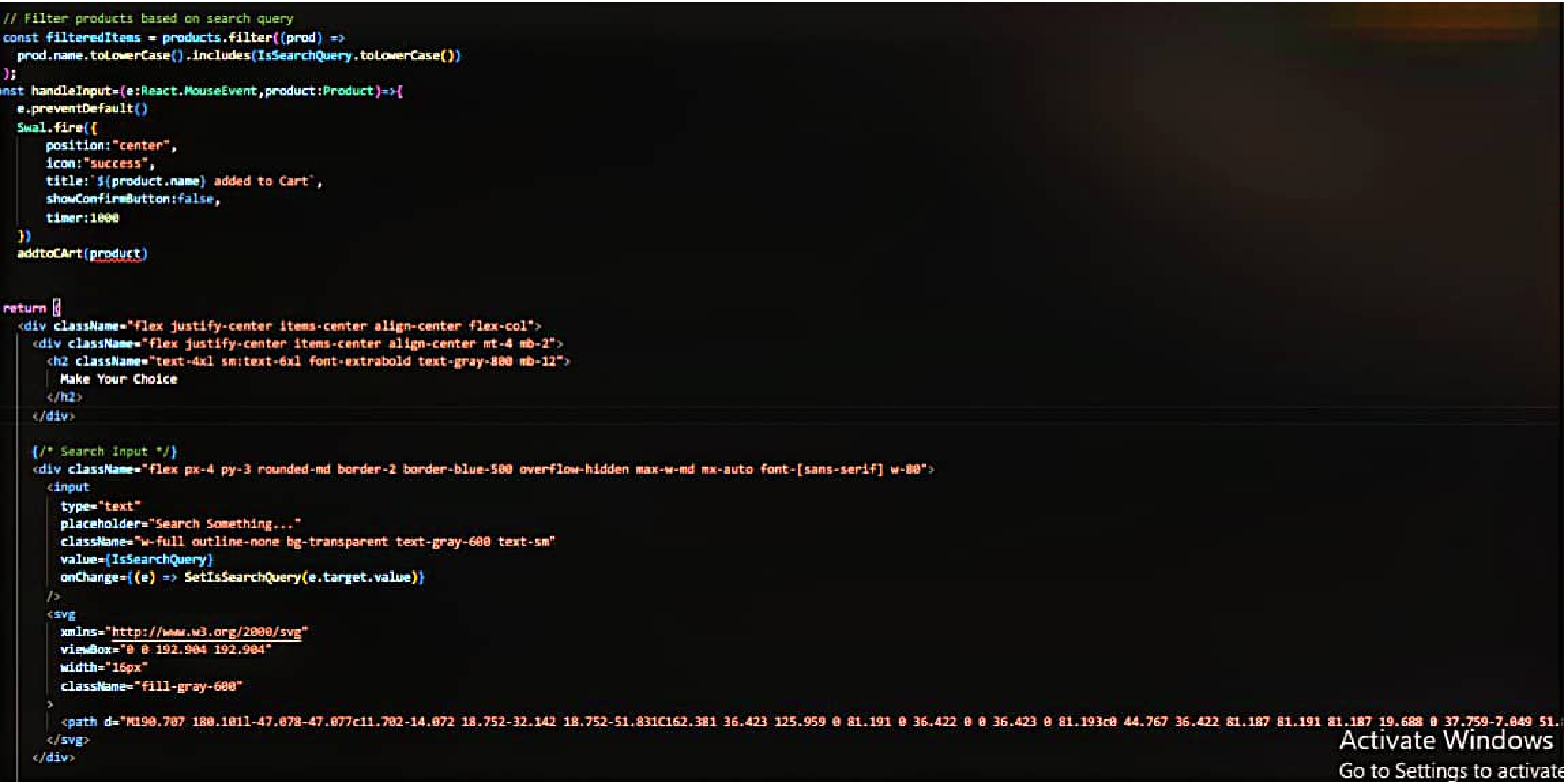
```
    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageUrl);

    if (imageId) {
      const document = {
        _type: 'products',
        name: product.name,
        description: product.description,
        price: product.price,
        image: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        category: product.category,
        discountPercent: product.discountPercent,
        isNew: product.isNew,
        colors: product.colors,
        sizes: product.sizes
      };
```



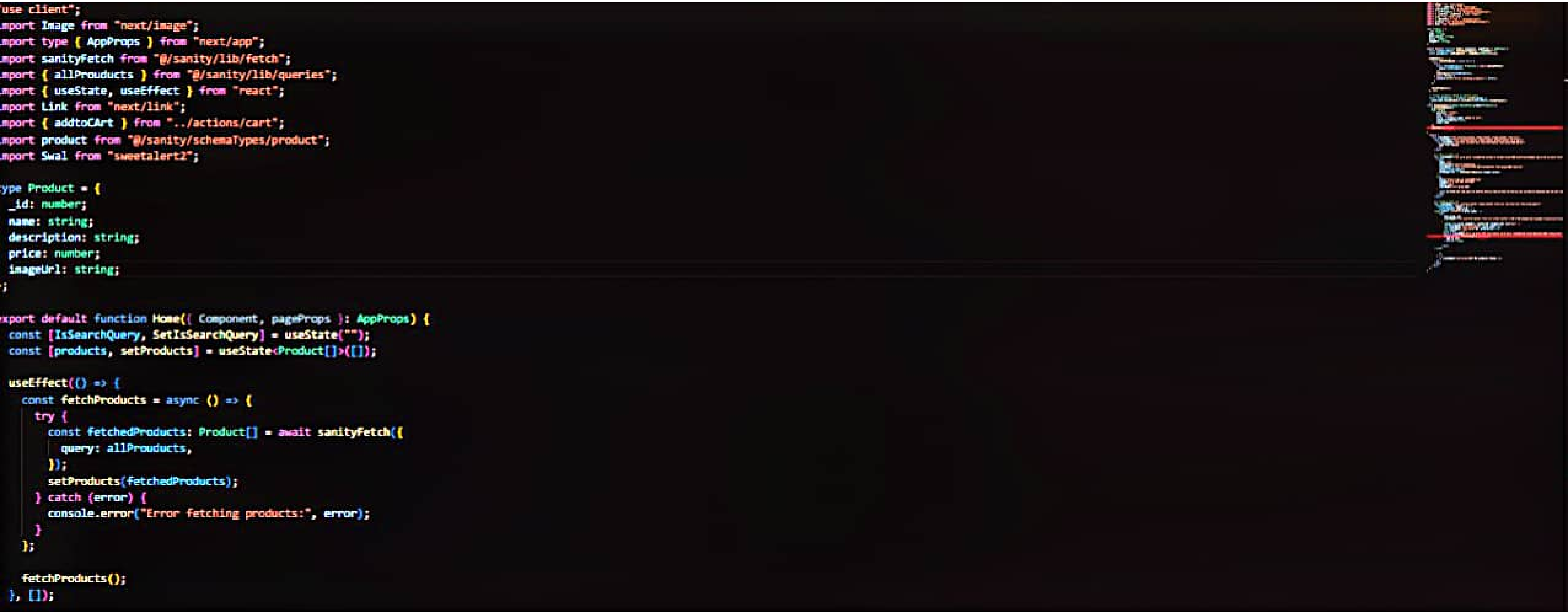
```
mport { createClient } from '@sanity/client';

onst client = createClient({
  projectId: 'xamv0099',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2025-02-06',
  token:
  'skK3HxEqjWu7lyUhTRxdTjw87oMSz0HjzcREQL9AC5QbVaJHDU0KqpPMzbqNfgAkbzxvxfzi1moojx9taRlLT74sAcFkFPtz7Wv6srTuZXdkRzCBOHL1e8WyvvnQVGuW4hRYbsvkOOG51
  ZtdLkQlxut58fFSdL7MglC376bKCMmEEm2JI9ck',
);

sync function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });
```



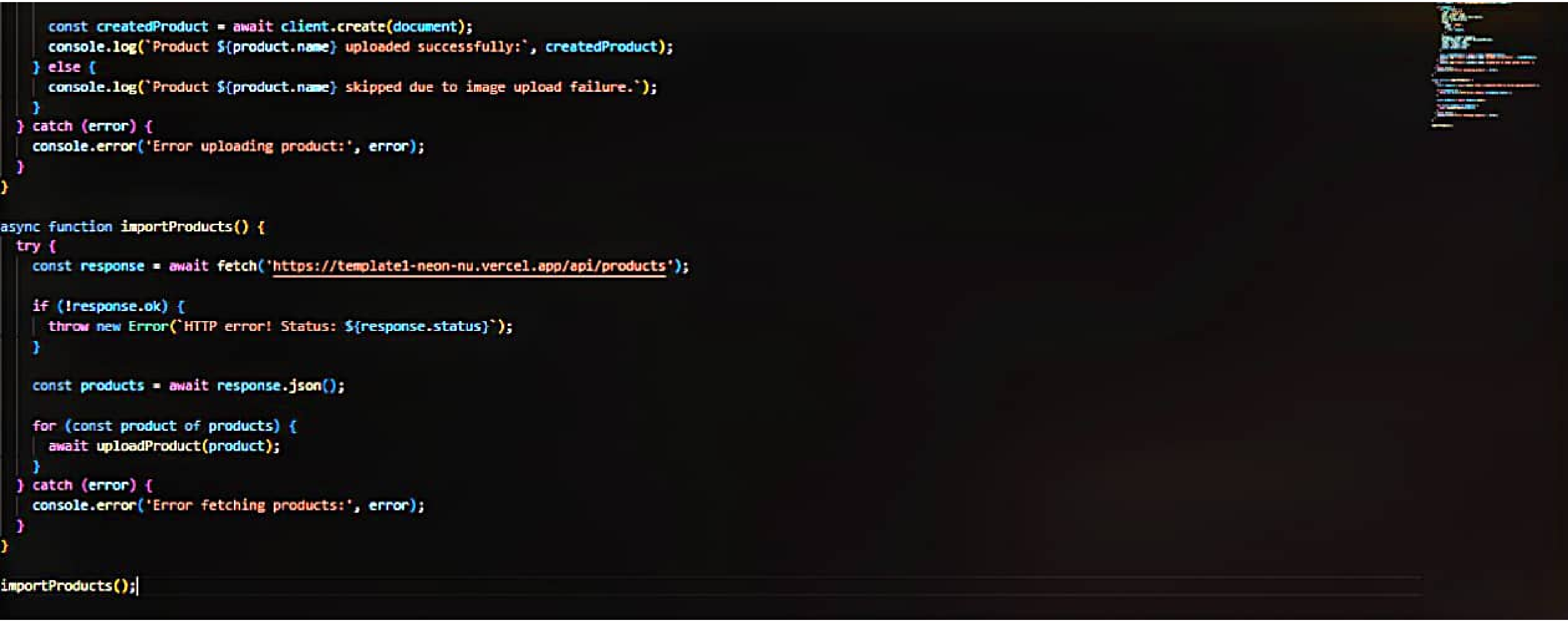
```
    {/* Product Grid */}
    <div className="flex justify-center items-center flex-col sm:flex-row flex-wrap gap-4">
      {filteredItems.length > 0 ? (
        filteredItems.map((prod) => (
    <Link href={`/products/${prod.name}` }>
        <div
            key={prod._id}
            className="flex justify-center flex-col items-center w-60 h-80 shadow-md rounded transition-transform transform hover:-translate-y-2"
          >
            <Image src={prod.imageUrl} width={80} height={60} alt="pic" />
            <h1 className="font-extrabold">{prod.name}</h1>
            <h4 className="font-bold">Rs {prod.price}</h4>
           <Link href="./cart">
           <button className="mt-2 bg-blue-400 text-white px-4 py-2 rounded-md hover:bg-blue-600 transition"
            onClick={(e)=>handleInput(e,product)}>
              Add to Cart
            </button></Link>

          </div>
    </Link>

        ))
      ) : (
        <p className="text-gray-600">No products found.</p>
      )}
    </div>
  </div>
);
```



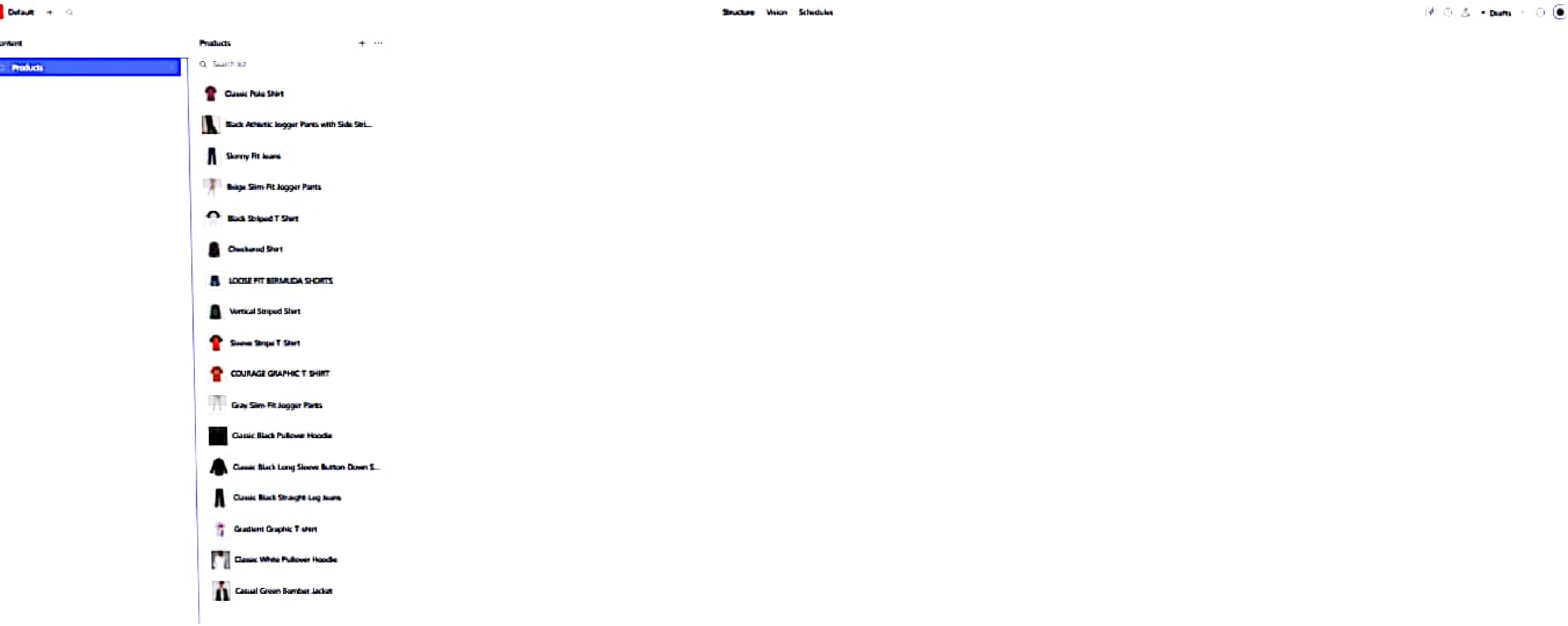
```
// Filter products based on search query
const filteredItems = products.filter((prod) =>
  prod.name.toLowerCase().includes(IsSearchQuery.toLowerCase())
);
nst handleInput=(e:React.MouseEvent,product:Product)=>{
  e.preventDefault()
  Swal.fire({
      position:"center",
      icon:"success",
      title:`${product.name} added to Cart`,
      showConfirmButton:false,
      timer:1000
  })
  addtoCArt(product)


return (
  <div className="flex justify-center items-center align-center flex-col">
    <div className="flex justify-center items-center align-center mt-4 mb-2">
      <h2 className="text-4xl sm:text-6xl font-extrabold text-gray-800 mb-12">
        Make Your Choice
      </h2>
    </div>

    {/* Search Input */}
    <div className="flex px-4 py-3 rounded-md border-2 border-blue-500 overflow-hidden max-w-md mx-auto font-[sans-serif] w-80">
      <input
        type="text"
        placeholder="Search Something..."
        className="w-full outline-none bg-transparent text-gray-600 text-sm"
        value={IsSearchQuery}
        onChange={(e) => SetIsSearchQuery(e.target.value)}
      />
      <svg
        xmlns="http://www.w3.org/2000/svg"
        viewBox="0 0 192.904 192.904"
        width="16px"
        className="fill-gray-600"
      >
        <path d="M190.707 180.101l-47.078-47.077c11.702-14.072 18.752-32.142 18.752-51.831C162.381 36.423 125.959 0 81.191 0 36.422 0 0 36.423 0 81.193c0 44.767 36.422 81.187 81.191 81.187 19.688 0 37.759-7.049 51.
      </svg>
    </div>
```



```
"use client";
mport Image from "next/image";
mport type { AppProps } from "next/app";
mport sanityFetch from "@/sanity/lib/fetch";
mport { allProuducts } from "@/sanity/lib/queries";
mport { useState, useEffect } from "react";
mport Link from "next/link";
mport { addtoCArt } from "../actions/cart";
mport product from "@/sanity/schemaTypes/product";
mport Swal from "sweetalert2";

ype Product = {
  _id: number;
  name: string;
  description: string;
  price: number;
  imageUrl: string;
;

xport default function Home({ Component, pageProps }: AppProps) {
  const [IsSearchQuery, SetIsSearchQuery] = useState("");
  const [products, setProducts] = useState<Product[]>([]);

  useEffect(() => {
    const fetchProducts = async () => {
      try {
        const fetchedProducts: Product[] = await sanityFetch({
          query: allProuducts,
        });
        setProducts(fetchedProducts);
      } catch (error) {
        console.error("Error fetching products:", error);
      }
    };

    fetchProducts();
  }, []);
```



```
      const createdProduct = await client.create(document);
      console.log(`Product ${product.name} uploaded successfully:`, createdProduct);
    } else {
      console.log(`Product ${product.name} skipped due to image upload failure.`);
    }
  } catch (error) {
    console.error('Error uploading product:', error);
  }
}

async function importProducts() {
  try {
    const response = await fetch('https://template1-neon-nu.vercel.app/api/products');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

importProducts();
```



Classic Polo Shirt

Black Athletic Jogger Pants with Side Stri...

Skinny Fit Jeans

Beige Slim Fit Jogger Pants

Black Striped T Shirt

Checkered Shirt

LOOSE FIT BERMUDA SHORTS

Vertical Striped Shirt

Sleeve Stripe T Shirt

COURAGE GRAPHIC T SHIRT

Gray Slim-Fit Jogger Pants

Classic Black Pullover Hoodie

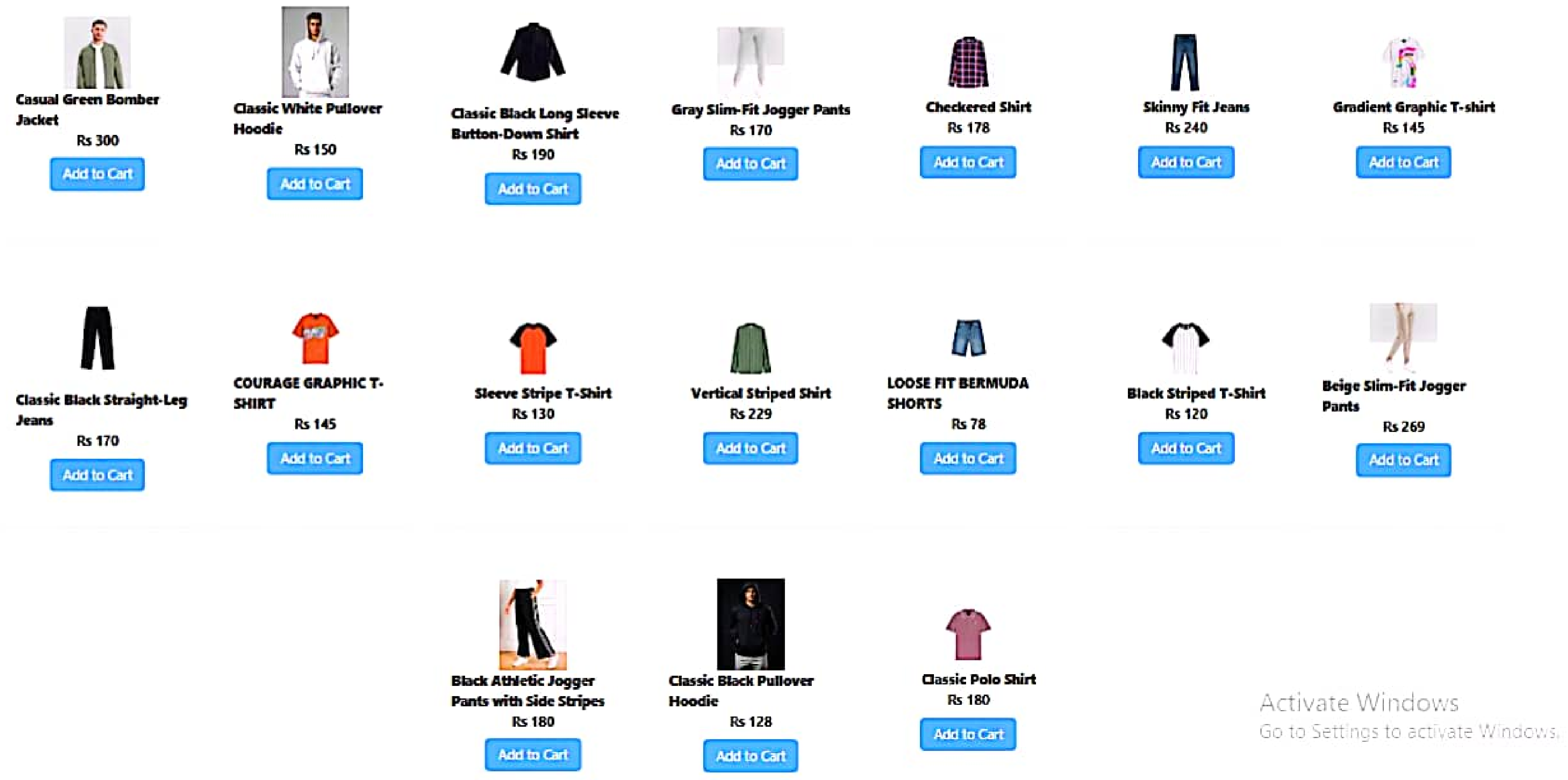
Classic Black Long Sleeve Button Down S...

Classic Black Straight Leg Jeans

Gradient Graphic T shirt

Classic White Pullover Hoodie

Casual Green Bomber Jacket



Casual Green Bomber Jacket
Rs 300
Add to Cart

Classic White Pullover Hoodie
Rs 150
Add to Cart

Classic Black Long Sleeve Button-Down Shirt
Rs 190
Add to Cart

Gray Slim-Fit Jogger Pants
Rs 170
Add to Cart

Checkered Shirt
Rs 178
Add to Cart

Skinny Fit Jeans
Rs 240
Add to Cart

Gradient Graphic T-shirt
Rs 145
Add to Cart

Classic Black Straight-Leg Jeans
Rs 170
Add to Cart

COURAGE GRAPHIC T-SHIRT
Rs 145
Add to Cart

Sleeve Stripe T-Shirt
Rs 130
Add to Cart

Vertical Striped Shirt
Rs 229
Add to Cart

LOOSE FIT BERMUDA SHORTS
Rs 78
Add to Cart

Black Striped T-Shirt
Rs 120
Add to Cart

Beige Slim-Fit Jogger Pants
Rs 269
Add to Cart

Black Athletic Jogger Pants with Side Stripes
Rs 180
Add to Cart

Classic Black Pullover Hoodie
Rs 128
Add to Cart

Classic Polo Shirt
Rs 180
Add to Cart